Retrieval of Relevant Opinion Sentences for New Products

Dae Hoon Park
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
dpark34@illinois.edu

Hyun Duk Kim
Twitter Inc.
1355 Market St Suite 900
San Francisco, CA 94103, USA
hkim@twitter.com

ChengXiang Zhai
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
czhai@cs.illinois.edu

Lifan Guo
TCL Research America
2870 Zanker Road
San Jose, CA 95134, USA
GuoLifan@tcl.com

ABSTRACT

With the rapid development of Internet and E-commerce, abundant product reviews have been written by consumers who bought the products. These reviews are very useful for consumers to optimize their purchasing decisions. However, since the reviews are all written by consumers who have bought and used a product, there are generally very few or even no reviews available for a new product or an unpopular product. We study the novel problem of retrieving relevant opinion sentences from the reviews of other products using specifications of a new or unpopular product as query. Our key idea is to leverage product specifications to assess product similarity between the query product and other products and extract relevant opinion sentences from the similar products where a consumer may find useful discussions. Then, we provide ranked opinion sentences for the query product that has no user-generated reviews. We first propose a popular summarization method and its modified version to solve the problem. Then, we propose our novel probabilistic methods. Experiment results show that the proposed methods can effectively retrieve useful opinion sentences for products that have no reviews.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Retrieval models*; H.4.0 [**Information Systems Applications**]: General

General Terms

Algorithms, Design

Keywords

opinion mining, probabilistic information retrieval

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

SIGIR'15, August 09 - 13, 2015, Santiago, Chile.
© 2015 ACM. ISBN 978-1-4503-3621-5/15/08 ...$15.00.
DOI: http://dx.doi.org/10.1145/2766462.2767748.

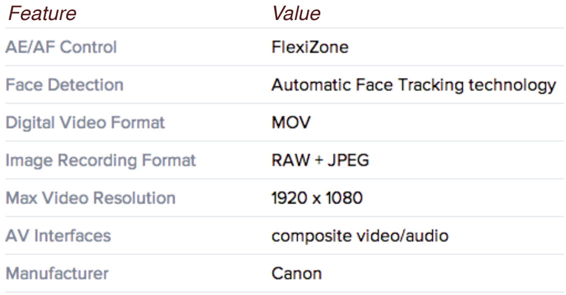
1. INTRODUCTION

The role of product reviews has been more and more important. Reevoo, a social commerce solutions provider, surveyed 1,000 consumers on shopping habits and found that 88 percent of them sometimes or always consult customer reviews before purchase.[1] According to the survey, 60 percent of them said that they were more likely to purchase from a site that has customer reviews on. Also, they considered customer reviews more influential (48%) than advertising (24%) or recommendations from sales assistants (22%). With the development of Internet and E-commerce, people's shopping habits have changed, and we need to take a closer look at it in order to provide the best shopping environment to consumers.

Even though product reviews are considered important to consumers, the majority of the products has only a few or no reviews. Products that are not released yet or newly released generally do not have enough reviews. Also, unpopular products in the market lack reviews because they are not sold and exposed to consumers enough. How can we help consumers who are interested in buying products with no reviews? In this paper, we propose methods to automatically retrieve review text for such products based on reviews of other products. Our key insight is that opinions on similar products may be applicable to the product that lacks reviews. For example, if products X and Y have the same CPU clock rate, then people's opinion on CPU clock rate for product X may be applicable to that for product Y as well. The similarity between products can be computed based on product specifications which are often available, where an example of product specifications is shown in Figure 1. %Here is an example of review text we manually retrieved for a certain product's specification "Resolution: 12.1 megapixels" from real reviews of products that have the same resolution.

> 12.1 MP captures very minute details even at highest zoom. This 12.1 megapixel megazoom offers an awesome value as the pictures it produces are on par with some cheap DSLRs. I will not longer bring my big DSR camera on my vacations. 12MP is too much, I use it with 8MP - that's more than plenty. What I like most about the W200 is my ability to get crystal clear 4000x3000 12.1 meg photos without having to

[1] https://www.reevoo.com/news/half-of-consumers-find-social-content-useful-when-shopping-online/



Feature	Value
AE/AF Control	FlexiZone
Face Detection	Automatic Face Tracking technology
Digital Video Format	MOV
Image Recording Format	RAW + JPEG
Max Video Resolution	1920 x 1080
AV Interfaces	composite video/audio
Manufacturer	Canon

Figure 1: A part of product specifications in CNET.com.

> spend a couple of thousand dollars on an digital SLR camera body and then even more cash on the accessories (e.g. lens).

Even though these sentences are not necessarily coherent opinions, they are clearly very useful for users to understand the product feature and get access to relevant discussion of opinions. Since a user would hardly have a clue about opinions on a new product, such a retrieved review text can be expected to be useful. As a minimum, it can be very useful to help users prioritizing what to read in the existing reviews of other products. Not only from a consumer's perspective, but also from a manufacturer's perspective, such techniques would be beneficial to collect opinions on its new or unpopular products. From the retrieved opinions, the manufacturers would be able to predict what consumers would think even before their product release and react to the predicted feedback in advance.

This paper makes the following contributions:

1. We introduce and study a novel problem of relevant opinion retrieval for products that do not have reviews in order to provide useful information to consumers and manufacturers. To the best of our knowledge, no previous work has addressed this problem.

2. To solve the problem, we propose a new probabilistic retrieval method, Translation model, Specifications Generation model, and Review and Specifications Generation model, as well as standard summarization model MEAD, its modified version MEAD-SIM, and standard ad-hoc retrieval method. Our suggested probabilistic methods are also able to retrieve per-feature opinions for a query product.

3. We create a new data set for evaluating the new problem and conduct experiments to show that our translation model indeed retrieves useful opinions and outperforms other baseline models. We also provide an interesting way to evaluate retrieved sentences for new products.

In order to evaluate the automatically retrieved opinions for a new or unpopular product, we pretend that the query product does not have reviews and predict its review text based on similar products. Then, we compare the predicted review text with the query product's actual reviews to evaluate the performance of suggested methods. Experiment results show that our translation model effectively retrieves opinions for a product without reviews and it significantly outperforms baseline methods.

2. RELATED WORKS

Reviews are one of the most popular sources in opinion analysis. Opinion retrieval and summarization techniques attracted a lot of attentions because of its usefulness in Web 2.0 environment. There are several surveys which summarize the existing opinion mining work [9, 21, 14]. Compared to text data in other general retrieval problems, opinionated articles such as product reviews have some different characteristics. In opinion analysis, analyzing polarities of input opinions are crucial. Also, majority of the opinion retrieval works are based on product feature (aspect) analysis. They first find sub-topics (features) of a target and show positive and negative opinions for each aspect. By further segmenting the input texts into the smaller units, they showed more details in a structured way [7, 15, 16, 18, 25, 28, 10]. Meanwhile, product reviews have been also employed to predict ratings [20, 5] or sales [3] of a product. However, no existing work addressed the problem of retrieving opinion sentences for new products yet.

In this paper, we also utilize unique characteristics of product data: specifications (structured data) as well as reviews (unstructured data). Although product specifications have been provided in many e-commerce web sites, there are only a limited number of studies that utilized specifications for product review analysis. Zhou and Chaovalit [32] performed sentiment classification on reviews using domain ontology database, which may be regarded as product specifications. Bhattattacharya *et al.* [2] employed IMDb's structured data to categorize documents, and Yu *et al.* [30] built an aspect hierarchy using product specifications and reviews. Wang *et al.* [29] and Peñalver-Martínez *et al.* [23] also employed product specifications to summarize product features. Product reviews and specifications were jointly modeled using topic models by Duan et al. [4] to improve product search and by Park et al. [22] to generate augmented specifications with useful information. Park et al. [22] retrieved review sentences for each (feature, value) pair, but they did not study their model's performance on products with no reviews. In addition, their model does not consider similarity among products or specifications, which is an important factor for the problem. Likewise, there are a few studies that employed product specifications, but their goals are different from ours.

Our work is related to text summarization, which considers centrality of text. Automatic text summarization techniques have been studied for a long time due to the need of handling large amount of electronic text data [19, 11, 6]. Automatic summarization techniques can be categorized into two types, extractive summary and abstractive summary. Extractive summarization makes a summary by selecting representative text segments, usually sentences, from the original documents. Abstractive summarization does not directly reuse the existing sentences but generates sentences based on text analysis. Our work is similar to extractive summarization in that we select sentences from original documents but different in that we retrieve sentences for an entity that does not have any text. Among the previous work, MEAD [26] is one of the most popular public extractive summarization toolkits, which supports multi-document summarization in general domain. The goal of MEAD is different from ours in that we want a summary for a specific product, and also MEAD does not utilize external structured data (specifications).

Cold start problem in recommendation systems [27], where no one has rated new items yet, is also related to our problem. However, unlike rating connections between items and users, each user review carries its unique and complex meaning, which makes the problem more challenging. Moreover, our goal is to provide useful relevant opinions about a product, not recommending a product. XML retrieval [12] that utilizes structured information of documents is also related to our work, in that reviews and specifications can be represented as a special XML. However, unlike general XML retrieval, in this paper, we propose more specialized methods for product reviews using product category and specifications. In addition, because we require the retrieved sentences to be central in reviews, we consider both centrality and relevance while general retrieval methods focus on relevance only. As far as we know, none of the existing work tried to solve the same problem as ours.

3. PROBLEM DEFINITION

The product data consists of N products $\{P_1, ..., P_N\}$. Each product P_i consists of its set of reviews $R_i = \{r_1, ..., r_m\}$ and its set of specifications $S_i = \{s_{i,1}, ..., s_{i,F}\}$, where a specification $s_{i,k}$ is a feature-value pair, $(f_k, v_{i,k})$, and F is the number of features. Given a query product P_z, for which R_z is not available, our goal is to retrieve a sequence of relevant opinion sentences T in K words for P_z.

Note that our problem setup is a mixture of retrieval and summarization. On the one hand, it can be regarded as a ranking problem, similar to retrieval; on the other hand, it can also be regarded as a summarization problem since we restrict the total number of words in the retrieved opinions .

This is a new problem that has not been addressed in any previous work. The problem is challenging for several reasons. Retrieved sentences for P_z should conform to its specifications S_z while we do not know which sentences are about which specific feature-value pair. In addition, the retrieved sentences should be central across relevant reviews so that they reflect central opinions. Despite the challenges, we try to show that achieving the goal is feasible. In the next a few sections, we propose multiple methods to solve the problem.

4. OVERALL APPROACH

When reviews are not available for a product, a consumer has no way to obtain opinions on it. In order to help consumers in such situation, we believe that product specifications are the most valuable source to find similar products. We thus leverage product specifications to find similar products and choose relevant sentences from their user reviews. In this approach, we assume that if products have similar specifications, the reviews are similar as well. For example, here is an actual review sentence from the review of a digital camera that takes a picture at high resolution: "the best camera I have ever owned, takes unbelievable crisp sharp photos with it's 16.1 Megapixels." It is admitted that the consumer is very impressed with the feature-value pair, ("Resolution", "16.1 Megapixels"), and we can expect that other digital cameras with the same feature-value pair could impress their consumers as well. The assumption may not be valid in some cases, *i.e.*, same specifications may yield very different user reviews. We thus try to retrieve "central" opinions from similar products so that the retrieved sentences can become clearly useful.

5. SIMILARITY BETWEEN PRODUCTS

We assume that similar products have similar feature-value pairs (specifications). In general, there are many ways to define a similarity function. We are interested in finding how well a basic similarity function will work although our framework can obviously accommodate any other similarity functions. Therefore, we simply define the similarity function between products as

$$SIM_p(P_i, P_j) = \frac{\sum_{k=1}^{F} w_k SIM_f(s_{i,k}, s_{j,k})}{\sum_{k=1}^{F} w_k} \tag{1}$$

where w_k is a weight for the feature f_k, and the weights $\{w_1, ..., w_F\}$ are assumed identical ($w_k = 1$) in this study, so the similarity function becomes

$$SIM_p(P_i, P_j) = \frac{\sum_{k=1}^{F} SIM_f(s_{i,k}, s_{j,k})}{F} \tag{2}$$

where $SIM_f(s_{i,k}, s_{j,k})$ is a cosine similarity for feature f_k between P_i and P_j and is defined as

$$SIM_f(s_{i,k}, s_{j,k}) = \frac{\mathbf{v_{i,k}} \cdot \mathbf{v_{j,k}}}{\sqrt{\sum_{v \in \mathbf{v_{i,k}}} v^2} \sqrt{\sum_{v \in \mathbf{v_{j,k}}} v^2}} \tag{3}$$

where $\mathbf{v_{i,k}}$ and $\mathbf{v_{j,k}}$ are phrase vectors in values $v_{i,k}$ and $v_{j,k}$, respectively. Both $SIM_p(P_i, P_j)$ and $SIM_f(s_{i,k}, s_{j,k})$ range from 0 to 1.

In this paper, we define the phrases as comma-delimited feature values. $SIM_f(s_{i,k}, s_{j,k})$ is similar to cosine similarity function, which is used often for measuring document similarity in Information Retrieval (IR), but the difference is that we use a phrase as a basic unit while a word unit is usually adopted in IR. We use a phrase as a basic unit because majority of the words may overlap in two very different feature values. For example, the specification phrases "Memory Stick Duo", "Memory Stick PRO-HG Duo", "Memory Stick PRO Duo", and "Memory Stick PRO Duo Mark2" have high word cosine similarities among themselves since they at least have 3 common words while the performances of the specifications are very different. Thus, our similarity function with phrase unit counts a match only if the phrases are the same.

6. METHODS

In this section, we suggest multiple methods for relevant opinion sentences retrieval. We first suggest a standard summarization tool, MEAD [26]. In order to make up for the MEAD's weak points, we also suggest modified version of MEAD. Then, we propose our probabilistic models to solve the problem.

6.1 MEAD: Retrieval by Centroid

For our problem, text retrieval based only on query-relevance is not desirable. The retrieved sentences need to be central in other reviews in order to obtain central opinions about specifications. For example, if there are more opinions that contains a word "big" than a word "small" for a certain feature-value pair, it is desired to assign higher score to the sentences having the word "big". However, since the query contains only feature-value pair words, classic information retrieval approaches are not able to prefer such sentences. Therefore, we suggest using a method that considers centrality among sentences.

MEAD [26] is a popular centroid-based summarization tool for multiple documents, and it was shown to effectively generate summaries from a large corpus. It provides an auto-generated summary for multiple documents. For a corpus R, a score of ith sentence t in a document is computed by sum of centroid and position scores of words, which is defined as

$$score(t; R) = w_c C_t + w_o O_t \tag{4}$$

where C_t is a sum of centroid scores of words in t, which is defined as $C_t = \sum_w C_{w,t}$, and O_t is a position score, which gives higher score to the sentences appearing earlier in a document and defined as $O_t = \frac{(n-i+1)}{n} \cdot C_{max}$ where n is the number of sentences in the document and C_{max} is the maximum centroid score in the document. Centroid score of a word, $C_{w,t}$, is a TFIDF value in the corpus R, and w_c and w_o are weights for C_t and O_t, respectively. Please refer to [26] for more details.

In order to retrieve sentences that are likely to be relevant to the query product P_z, which has no reviews, we employ specifications to find products similar to P_z and use the similarity as a clue for finding relevant sentences. Since the score formula (4) utilizes only centrality and does not consider relevance to the query product, we augment it with product similarity to P_z so that we can find sentences that are query-relevant and central at the same time. In addition, MEAD employs position score that is reasonable for news articles, but it may not be appropriate for reviews; unlike news articles, it is hard to say that the sentences appearing earlier in the reviews are more important than those appearing later. Thus, we remove position score term from formula (4), and we augment it with similarity to query. The new score function is defined as

$$score(t, S_y; R, S_z) = C_t \cdot SIM_p(S_y, S_z) \tag{5}$$

where t is a sentence in a review for product P_y and SIM_p (S_y, S_z) is a product similarity between P_y and the query product P_z, which is defined in equation (2).

6.2 Probabilistic Retrieval

To solve the problem in a more principled way, we introduce our probabilistic methods. Query likelihood retrieval model [1], which assumes that a document generates a query, has been shown to work well for ad-hoc information retrieval. Similarly, we attempt to generate the query specifications S_z from a candidate sentence t via several generative scenarios.

6.2.1 Specifications Generation Model

The generative story is described as follows. Each sentence t from reviews of its product P_y first generates its specifications S_y. The specifications S_y then generates the query specifications S_z. Following the dependencies among variables, the scoring function is defined as

$$\begin{aligned} score(t, S_y; R, S_z) &\propto p(t, S_y|S_z) \\ &= \frac{p(S_z|S_y)p(S_y|t)p(t)}{p(S_z)} \end{aligned} \tag{6}$$

We can interpret $p(t, S_y|S_z)$ as the probability that t and S_y satisfy information needs of a user given S_z. $p(S_z|S_y)$ measures proximity of S_y to S_z. $p(S_y|t)$ measures proximity of t to S_y, and $p(t)$ is a general preference on t. Since we assume no preference on sentences, we ignore $p(t)$ for ranking. $p(S_z)$ is also ignored because it does not affect the ranking of sentences for S_z. Thus, the formula assigns high score to a sentence if its specifications S_y match S_z well and the sentence t matches its specifications S_y well. $p(t, S_y|S_z)$ is then defined as

$$\begin{aligned} p(t, S_y|S_z) &\propto p(S_z|S_y)p(S_y|t) \\ &= \sum_{k=1}^{F} p(s_{z,k}|s_{y,k})p(s_{y,k}|t) \end{aligned} \tag{7}$$

where a set of specifications such as S_y is decomposed into feature-value pairs $s_{y,k}$. We assume that a k'th feature-value pair of one specification set generates only the k'th feature-value pair of another specification set, not other feature-value pairs. This is to ensure that sentences not related to a specification $s_{z,k}$ are scored low even if their word score $p(s_{y,k}|t)$ is high. $p(s_{z,k}|s_{y,k})$, proximity of $s_{y,k}$ to $s_{z,k}$, is estimated as follows.

$$p(s_{z,k}|s_{y,k}) \propto \frac{SIM_f(s_{z,k}, s_{y,k})}{\sum_{s \in Distinct(k)} SIM_f(s, s_{y,k})} \tag{8}$$

where $Distinct(k)$ is a set of distinct feature-value pairs for a feature f_k. $p(s_{y,k}|t)$ is defined as

$$p(s_{y,k}|t) = \prod_{w \in s_{y,k}} p(w|t) = \prod_{w \in U} p(w|t)^{c(w,s_{y,k})} \tag{9}$$

where U is a vocabulary set in corpus R, and $c(w, s_{y,k})$ is a count of word w in the feature-value pair $s_{y,k}$. $p(w|t)$ follows t's unigram language model [24], and it means a word w's likelihood in a sentence t. One of the standard ways to estimate $p(w|t)$ is using maximum likelihood (ML) estimator, which gives $p(w|t) = \frac{c(w,t)}{|t|}$, where $c(w, t)$ is the count of w in t, and $|t|$ is the number of words in t. Thus, $p(s_{y,k}|t)$, likelihood of a feature-value pair $s_{y,k}$ in a sentence t, becomes higher if more words in the feature-value pair appear often in t. To avoid over-fitting and prevent $p(s_{y,k}|t)$ from being zero, we smooth $p(w|t)$ with Jelinek-Mercer smoothing method [8], which is shown in [31] to work reasonably well. Using Jelinek-Mercer smoothing, $p(w|t)$ is defined as:

$$p(w|t) = (1 - \lambda)p_{ml}(w|t) + \lambda p(w|R) \tag{10}$$

where $p_{ml}(w|t)$ and $p(w|R)$ follow a sentence language model and a corpus language model, respectively, estimated with ML estimator. To smooth $p(w|t)$, a reference language model $p(w|R)$ is used so that we can have general word likelihood that nicely augments $p_{ml}(w|t)$. The resulting $p(w|t)$ can be regarded as weighted average of $p_{ml}(w|t)$ and $p(w|R)$.

6.2.2 Review and Specifications Generation Model

Specifications Generation model in section 6.2.1 does not consider centrality among reviews. However, as explained in section 6.1, centrality as well as query-relevance should be considered for the task. Here, we assume that a candidate sentence t of product P_y generates the product's reviews R_y except itself t. This generation enables us to measure centrality of t among all other sentences in the reviews for P_y. Then, t and $R_y^{\setminus t}$ jointly generate its specifications S_y, where $R_y^{\setminus t}$ is a set of reviews for P_y except the sentence t. Intuitively, it makes more sense for S_y to be generated by both t and $R_y^{\setminus t}$ than by only t. S_y then generates the query specifications S_z. Following the dependencies, the score function is defined as

$$\begin{aligned} score(t, R_y^{\setminus t}, S_y; R, S_z) &\propto p(t, R_y^{\setminus t}, S_y|S_z) \\ &= \frac{p(S_z|S_y)p(S_y|t, R_y^{\setminus t})p(R_y^{\setminus t}|t)p(t)}{p(S_z)} \\ &\propto p(S_z|S_y)p(S_y|t, R_y^{\setminus t})p(R_y^{\setminus t}|t) \end{aligned} \tag{11}$$

where $p(t)$ and $p(S_z)$ are ignored for the same reason as in section 6.2.1. Now, $p(R_y^{\setminus t}|t)$, a proximity of t to the reviews $R_y^{\setminus t}$, is computed to consider centrality of t. Also, $p(S_y|t, R_y^{\setminus t})$, a proximity of t and $R_y^{\setminus t}$ to the specifications S_y, is computed to promote sentences from reviews that match its specifications well. Thus, a sentence t is preferred if (1) its specifications S_y is similar to S_z, (2) S_y represent its reviews R_y well, and (3) $R_y^{\setminus t}$ represents t well.

$p(t, R_y^{\backslash t}, S_y|S_z)$ can be re-written as

$$p(t, R_y^{\backslash t}, S_y|S_z) = p(R_y^{\backslash t}|t) \sum_{k=1}^{F} p(s_{z,k}|s_{y,k}) \prod_{w \in s_{y,k}} p(w|t, R_y^{\backslash t}) \tag{12}$$

where $p(w|t, R_y^{\backslash t})$ is smoothed to $(1-\lambda)\delta(w|t, R_y^{\backslash t}) + \lambda p(w|R)$, where $\delta(w|t, R_y^{\backslash t})$ is defined as

$$\delta(w|t, R_i) = \begin{cases} 0 & \text{if } w \notin t \\ \frac{c(w,t)+c(w,R_i)}{|t|+|R_i|} & \text{if } w \in t \end{cases} \tag{13}$$

We ignore w if w is not in t in order to require the retrieved sentences to contain words in $s_{y,k}$. The proximity of t to $R_y^{\backslash t}$, $p(R_y^{\backslash t}|t)$, is estimated by TFIDF cosine similarity function $SIM(R_y^{\backslash t}, t)$, where TFIDF cosine similarity between documents d and d' is defined as

$$SIM(d, d') = \frac{\sum_{w \in d, d'} c(w, d) \cdot c(w, d') \cdot IDF(w)^2}{\sqrt{\sum_{w \in d} (c(w, d) \cdot IDF(w))^2} \cdot \sqrt{\sum_{w' \in d'} (c(w', d') \cdot IDF(w'))^2}} \tag{14}$$

where IDF of word w is defined as

$$IDF(w) = \log \frac{|R|}{1 + DF(w)} \tag{15}$$

where $|R|$ is the number of reviews in the whole corpus, and $DF(w)$ is the number of documents that contain w.

6.2.3 Translation Model

In Review and Specifications Generation model, we assumed a sentence t of product P_y generates its reviews R_y, and t and R_y jointly generate their specifications S_y. However, we can also assume that t generates reviews of an arbitrary product because there may be better reviews that can represent t and generate S_y well. In other words, there may be a product P_x that translates t and generates P_z based on the translation with a better performance.

The generative story is described as follows. A candidate sentence t of a product P_y generates each review set of all products, which will be used as translations of t. t and each of the generated review sets, R_x, jointly generates t's specifications S_y, and S_y generates specifications of R_x, S_x, and the query specifications S_z. We intend S_y to generate specifications of the translating product S_x so as to penalize the translating product if its specifications are not similar to S_y. Following the generative story, the score function is defined as

$$\begin{aligned} &score(t, S_y; R, S_z) \propto p(t, S_y|S_z) \\ &= \frac{p(S_z|S_y) \sum_{P_x \in P^{\backslash z}} p(S_x|S_y) p(S_y|t, R_x) p(R_x|t) p(t)}{p(S_z)} \\ &\propto p(S_z|S_y) \sum_{P_x \in P^{\backslash z}} p(S_x|S_y) p(S_y|t, R_x) p(R_x|t) \end{aligned} \tag{16}$$

where $p(S_z)$ and $p(t)$ are ignored for the same reason as before. As described, the score function contains a loop over all products (except P_z), instead of using only t's review set R_y, to get the votes from all translating products. The features in different specifications are paired together, which decompose $p(t, S_y|S_z)$ as follows.

$$\begin{aligned} &p(t, S_y|S_z) \\ &\propto \sum_{k=1}^{F} p(s_{z,k}|s_{y,k}) \sum_{P_x \in P^{\backslash z}} p(s_{x,k}|s_{y,k}) p(s_{y,k}|t, R_x) p(R_x|t) \\ &= \sum_{k=1}^{F} p(s_{z,k}|s_{y,k}) \sum_{P_x \in P^{\backslash z}} p(s_{x,k}|s_{y,k}) p(R_x|t) \prod_{w \in s_{y,k}} p(w|t, R_x) \end{aligned} \tag{17}$$

where proximity between specifications are estimated using cosine similarity function SIM_f as in specifications generation model, and the proximity of t to arbitrary reviews R_x, $p(R_x|t)$, is estimated by TFIDF cosine similarity function. In order to consider the case P_y is the same as P_x, we define $p(w|t, R_x)$ as

$$p(w|t, R_x) = \begin{cases} (1-\lambda)\delta(w|t, R_x) + \lambda p(w|R) & \text{if } P_x \neq P_y \\ (1-\lambda)\delta(w|t, R_x^{\backslash t}) + \lambda p(w|R) & \text{if } P_x = P_y \end{cases} \tag{18}$$

Meanwhile, looping over all non-query products is probably too expensive in terms of computational complexity. We thus choose X translating products P^X to reduce the complexity. Perhaps, the most promising translating products may be those who are similar to the query product P_z. We want the retrieved sentences to be translated well by the actual reviews of P_z, which means that those reviews of products not similar to P_z are not considered important. Since we assume that products similar to P_z are likely to have similar reviews, we exploit the similar products' reviews to approximate R_z, where we measure similarity using specifications. Therefore, we loop over only X translating products P^X that are most similar to P_z, where similarity function SIM_p is employed to measure similarity between products. Since P_x needs to be similar to P_z, we further assume that P_x generates P_z, which yields proximity of P_x to P_z, $p(P_z|P_x)$, and it is defined as

$$p(P_z|P_x) = \frac{SIM_p(P_z, P_x)}{\sum_{x' \in P^X} SIM_p(P_z, P_{x'})} \tag{19}$$

and this product-level similarity is used as a weight of P_x in formula (17).

7. EXPERIMENTAL SETUP

7.1 Data Set

Since we study a new task that has not been studied before, there is no existing test collection available to use for evaluation. We thus must solve the challenge of creating a test set. We address this problem by using products with known reviews as test cases. We pretend that we do not know their reviews and use our methods to retrieve sentences in K words; we then compare these results with the actual known review of a test product. This allows for evaluating the task without requiring manual work, and is a reasonable way to perform evaluation because it would reward a system that can retrieve review sentences that are very similar to the actual review sentences of a product.

We now describe how to build our data set in detail. First, it is required for our problem to obtain reviews and specifications for products, and this kind of data is available in several web sites such as Amazon.com, BestBuy.com, and CNET.com. Among them, we chose CNET.com because they have reasonable amount of review data and relatively well-organized specifications. There are several product categories in CNET.com, and we chose digital camera and MP3 player categories since they are reasonably popular and therefore the experiment results can yield significant impact. From CNET.com, we crawled product information for all products that were available on February 22, 2012 in both categories. For each product, we collected its user reviews and specifications.

Table 1: Statistics of the data for digital camera and MP3 player categories.

	Digital Camera	MP3 Player
Num. of products	1,153	605
Num. of reviews	12,779	14,159
Num. of sentences	137,599	291,858
Num. of word tokens	754,888	172,5192
Vocabulary size	6442	6959
Num. of features	9	8
Num. of distinct feature values	1,038	384

We pruned out products that do not contain reviews or specifications. (We found that about two thirds of the products didn't have any user reviews.) To preprocess the review text, we performed sentence segmentation, word tokenization, and lemmatization using Stanford CoreNLP [17] version 1.3.5. We lowered word tokens and removed punctuations. Then, we removed word tokens that appear in less than five reviews and stopwords. We also preprocessed specifications data. In general, specifications contain dozens or hundreds of distinct features, and many of them are not mentioned in the reviews. Therefore, we choose features that are considered important by users. In order to choose such key features, we simply adopt highlighted features provided by CNET.com assuming that they chose the features based on importance. The highlighted features are listed in Table 2. We removed feature values that appear in less than five products. Then, we tokenized the feature and feature value words, and we lowered the word tokens. The statistics of the reviews and specifications data is shown in Table 1. While digital camera category has more products, more reviews are written for mp3 player categories. Also, in general, users wrote more texts per review for mp3 players than digital cameras. The number of highlighted features used for digital cameras is similar to that for mp3 players while there are much more distinct feature values for digital cameras.

Table 2: CNET.com's highlighted features for digital camera and MP3 player categories.

Digital Camera	MP3 Player
Manufacturer	Manufacturer
Product Type	Product Type
Resolution	Digital Storage
Digital Video Format	Flash Memory Installed
Image Stabilizer	Built-in Display – Diagonal Size
Lens System – Type	Battery / Power – Battery
Memory / Storage – Supported Mem. Cards	Digital Player / Recorder – Supported Digital Audio Standards
Camera Flash – Camera Flash	Battery / Power – Mfr. Estimated Battery Life
Optical Sensor Type	

In order to evaluate the performance of our methods for retrieving review sentences for a new or unpopular product, we perform the following experiment. To choose test products, which will be regarded as products with no reviews, we selected top 50 qualified products by the number of reviews in each category in order to obtain statistically reliable gold standard data. Please note that we did not select (qualify) products that have their different versions such as colors or editions, in order to ensure that review sentences from the different version of the same product are not retrieved. For each of the top products, P_z, all sentences of other products are regarded as candidate sentences. Pretending P_z does not have any reviews, we rank those candidate sentences and generate a text of first K word tokens, and we compare it with the actual reviews of P_z. We assume that if the generated review text is similar to the actual reviews, it is a good review text for P_z. The average number of reviews in the top 50 products is 78.5 and 152.2 for digital cameras and mp3 players, respectively. For the probabilistic retrieval models, we use λ to control the amount of smoothing for language models, and we empirically set it to 0.5 for both product categories, which showed the best performance.

7.2 Evaluation Metrics

To evaluate a quality of the length-K retrieved text based on actual reviews for the query, we face another challenge: how should we measure the performance? We could consider using standard retrieval measures, but neither NDCG, nor MAP seems appropriate since we do not have multiple levels of judgments or even binary judgments. We thus decided to measure the proximity between the retrieved text and the actual reviews. Regarding the retrieved text as a summary for the query product, we can view our task as similar to multiple document summarization, whose goal is to generate a summary of multiple documents. Thus, we employ ROUGE evaluation method [13], which is a standard evaluation system for multiple document summarization. In general, ROUGE evaluates the quality of an automatically generated summary by comparing it with one or more manually generated reference summaries. Assuming the actual reviews of the query product are manually generated reference summaries, we can adopt ROUGE to evaluate the retrieved sentences. Among various ROUGE metrics, we employ ROUGE-1 and ROUGE-2, which are unigram and bigram matching metrics, respectively, and have been shown to perform well for the task. We compute precision, recall, and F1-score of each metric. For example, recall of ROUGE-n is defined as

$$\text{ROUGE-}n(r,s) = \frac{\sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{gram_n \in S} Count(gram_n)} \quad (20)$$

where r and s are reference and retrieved summaries, respectively, $gram_n$ is n-gram text, $Count_{match}(gram_n)$ is the maximum number of n-grams co-occurring in the retrieved summary and a reference summary. When there are multiple reference summaries are available, they use the following evaluation formula.

$$\text{ROUGE-}n_{multi} = max_i \text{ROUGE-}n(r_i, s) \quad (21)$$

Please note that each of the precision, recall, and F1-score takes the maximum from the reference summaries. More details about ROUGE can be found in [13].

However, the problem of ROUGE metrics is that it does not consider importance of words. All words have different level of importance; for example a word such as "of" is much less important than a word "megapixel" since "of" appears too often in documents and does not carry useful information. If a retrieved text contains many unimportant words, it may obtain a high score by ROUGE metrics, which is not desired. Therefore, we also employ TFIDF cosine similarity, which considers word importance by Inverse Document Frequency (IDF). TFIDF cosine similarity function between two documents is defined in equation (14). While the formula measures similarity based on bag of words, bigram provides important information about distance among words, so we adopt bigram-based TFIDF cosine similarity as well. Similar to ROUGE-n_{multi}, we take a maximum from $SIM(r_i, s)$ among different reference summaries because we still evaluate based on multiple reference summaries. For both ROUGE and SIM metrics, we use retrieved text length 100, 200, and 400, which reflect diverse users' information needs.

8. EXPERIMENT RESULTS

8.1 Qualitative Analysis

Table 3: Top ten sentences retrieved for Pentax *ist DS (Digital Camera) by Translation model with X=5.

(1) This was my first and my last Pentax .
(2) This pentax is a great value for money , and a nice entry level dslr , compatible with most Pentax lens .
(3) I have found the Pentax DL to be high quality , with great features .
(4) Nice job pentax .
(5) I have been a Pentax SLR user for years , beginning with the SuperProgram , ZX-50 , and ZX-5n .
(6) When I bought it , I was in bankruptcy and the cheaper Pentax came to me .
(7) Pentax have been making great lenses and cameras for a long time , and this range is no exception .
(8) Great photos , color , ease of use , compact size , compatible with Pentax mount lenses .
(9) I had owned a great 35mm Pentax camera before that took wonderful pictures , which , after 20 years went caput .
(10) Very smart Pentax .

Table 4: Specifications for Pentax *ist DS. Note that some feature values are not available.

Feature	Value
Manufacturer	Pentax
Product Type	Digital camera - SLR
Resolution	6.1 megapixels
Digital Video Format	
Image Stabilizer	
Lens System – Type	3 x x Zoom lens - 18 mm - 55 mm - F/3.5-5.6 DA Pentax KAF
Memory / Storage – Supported Mem. Cards	SD Memory Card
Camera Flash – Camera Flash	Pop-up flash
Optical Sensor Type	CCD

In order to see the usefulness of the sentences retrieved by our novel Translation model, we show the top retrieved sentences for query products and compare them with the actual review sentences for the query products. Table 3 lists top retrieved sentences for a product in each category, where the sentences are ordered by their scores, and the specifications of the product is listed in Table 4.We set the number of translating products to five, which is reasonable if we consider the computational cost of the model.

For the digital camera Pentax *ist DS, several top retrieved sentences such as (2), and (8) mention about its compatibility with Pentax lenses. Surprisingly, there were several reviews for Pentax *ist DS that praise its lens compatibility, and here are two actual examples from review sentences: "Plus the DS is backwards compatible with all old Pentax lenses, which have a well-deserved reputation among photographers." and "I can use my pile of old (and very old) Pentax lenses including the m42 lenses." Also, the retrieved sentences such as (7), (8), (9), and possibly (3) mention about Pentax's great picture quality, which is supported by the following actual review sentences: "Amazingly sharp lens." and "It has a much better lens package than the Rebel and the base 20D kit." Sentences (2) and (6) claim the product's good value, which is again supported by actual review sentences: "Better value than you think" and "The camera is also cheaper than the comparable Nikon and Canon." The retrieved sentence such as (8) mentions about ease of use for the camera, and many users actually complimented the camera on its ease of use, indeed. The supporting sentences are as follows: "Very easy to use right out of the box." and "The controls are very easy to learn and are, for the most part, very intuitive." Meanwhile, the sentence (1) carries inconsistent opinion, which shows negative sentiment on Pentax camera. Nevertheless, in a user's perspective, who does not know much about Pentax *ist DS or other Pentax cameras, the listed information would be highly informative especially if the camera has no or few reviews. Although some of the retrieved sentences do not carry useful information, it is clear that some other retrieved sentences are indeed useful.

Our probabilistic retrieval models have a capability of retrieving relevant sentences for a specific feature. For each of the probabilistic models, we can assume that the number of features F is one so that the score functions compute only for one feature. Table 5 shows top retrieved sentences for the feature "Lens System – Type" of Pentax *ist DS. As found in the top sentences for the whole product in Table 3, we can easily find that all the sentences except (2) praise the lens compatibility of Pentax, indeed. In addition, all sentences except (1) praises high quality of its lens, which is coherent with the top sentences for the whole product. From the sentences, users can learn much about the given product's lens such as other consumers' general sentiment and specific reasons why they like or dislike its lens.

Table 5: Top sentences retrieved by Translation model (X=5) specifically for the feature "Lens System – Type" of Pentax *ist DS.

(1) This pentax is a great value for money , and a nice entry level dslr , compatible with most Pentax lens .
(2) Pentax have been making great lenses and cameras for a long time , and this range is no exception .
(3) Great photos , color , ease of use , compact size , compatible with Pentax mount lenses .
(4) The kit lens is better than what ships with some competitors , and the camera is compatible with most older Pentax lenses , making it possible to save hundreds by buying used lenses rather than having to sink money into new digital lenses .
(5) Compatibility with older Pentax lenses is a real bonus too , as these are usually of very high quality and can be picked up at good prices second-hand .

Manually finding relevant opinions for a query product or its specific feature is extremely time-consuming for users; they need to find similar products by manually comparing specifications and extract relevant and central sentences from all the reviews of the similar products, which may take too much time. Here, we verified the automatically retrieved sentences can be indeed useful for users. In the next section, we quantitatively compare our Translation model with other suggested methods.

8.2 Quantitative Evaluation

To retrieve review sentences that are likely to be written for a new or unpopular product, we employ several methods. In order to see the effectiveness of a standard ad-hoc retrieval method, we employ query likelihood (QL) language model approach [24], and we define the score func-

Table 6: Unigram and bigram TFIDF cosine similarity @ K for Digital Camera and MP3 Player categories.

Category	Model	COS1@100	COS1@200	COS1@400	COS2@100	COS2@200	COS2@400
Digital Camera	QL	0.112	0.128	0.147	0.0185	0.0215	0.0257
	MEAD	0.131	0.124	0.141	0.0258	0.0204	0.0184
	MEAD-SIM	0.136	0.130	0.158	0.0271	0.0226	0.0223
	SpecGen	0.143	0.173	0.206	0.0230	0.0270	0.0291
	ReviewSpecGen	0.171	0.208	0.231	0.0210	0.0244	0.0298
	Translation	**0.314**†‡	**0.327**†‡	**0.333**†‡	**0.0736**†‡	**0.0743**†‡	**0.0794**†‡
	(increase %)	(+131%)	(+152%)	(+111%)	(+172%)	(+229%)	(+256%)
MP3 Player	QL	0.090	0.99	0.118	0.0147	0.0159	0.0173
	MEAD	0.089	0.078	0.091	0.0123	0.0128	0.0117
	MEAD-SIM	0.131	0.136	0.145	0.0206	0.0197	0.0178
	SpecGen	0.153	0.183	0.208	0.0225	0.0274	0.0294
	ReviewSpecGen	0.206	0.227	0.253	0.0261	0.0270	0.0327
	Translation	**0.267**†‡	**0.297**†‡	**0.316**†‡	**0.0458**†‡	**0.0567**†‡	**0.0649**†‡
	(increase %)	(+104%)	(+118%)	(+118%)	(+104%)	(+188%)	(+265%)

tion as $score(t; R, S_z) = \sum_{k=1}^{F} \prod_{w \in s_{z,k}} p(w|t)$, where $p(w|t)$ is smoothed as in equation (10). We suggested a modified version of one of the standard summarization tools, MEAD-SIM in formula (5), which considers both query-relevance and centrality. We employ MEAD-SIM as one of the baseline methods, and we also show results from the basic MEAD in formula (4) to see the effect of query-relevance addition to MEAD; we set $w_c = 1$ and $w_o = 0$ since position score is inappropriate for reviews. We also introduced several probabilistic retrieval methods for the task. Review and Specifications Generation model (ReviewSpecGen) considers both query-relevance and centrality, so we use it as another baseline method. Specifications Generation model (SpecGen) focuses on query-relevance, and we show its results to compare with ReviewSpecGen and QL. We then suggested our novel Translation model (Translation). We tuned X to be 100 for digital cameras and 10 for mp3 players, unless otherwise specified, which showed the best TFIDF cosine similarity values. The results from Translation model are mainly compared with the two baselines MEAD-SIM and ReviewSpecGen. † and ‡ are used to mark if the improvement for Translation model is statistically (paired t-test with p=0.05) significant in each measure from MEAD-SIM and ReviewSpecGen, respectively. We also record how much Translation model outperforms MEAD-SIM in parentheses.

Table 6 shows TFIDF cosine similarity evaluation results for both digital cameras and mp3 players. Both unigram (COS1) and bigram (COS2) measures are listed for the suggested methods. In general, models that exploit specifications as query (MEAD-SIM, SpecGen, ReviewSpecGen, and Translation) except QL outperform MEAD, which does not compute query-relevance. QL outperforms MEAD in mp3 player data set, but it does not outperform other models in both data sets, since does not consider specifications similarity between products. MEAD-SIM outperforms MEAD in all cosine similarity measures (12/12), which means that centrality alone cannot perform well. ReviewSpecGen adds centrality computation to SpecGen, and the results show that its centrality helps it outperform SpecGen in most measures (9/12). ReviewSpecGen outperforms MEAD-SIM in all unigram measures (6/6) and most bigram measures (5/6). Translation model significantly outperforms MEAD-SIM in all measures (12/12), and the average performance increase percentage is 162%. It also significantly outperforms ReviewSpecGen in all measures (12/12), which means that choosing products similar to the query product as translating products was more effective than choosing only one product the candidate sentence belongs. Translation model outperforms other models especially in bigram measures, which means that Translation model retrieves more connected fragments that are in the actual reviews.

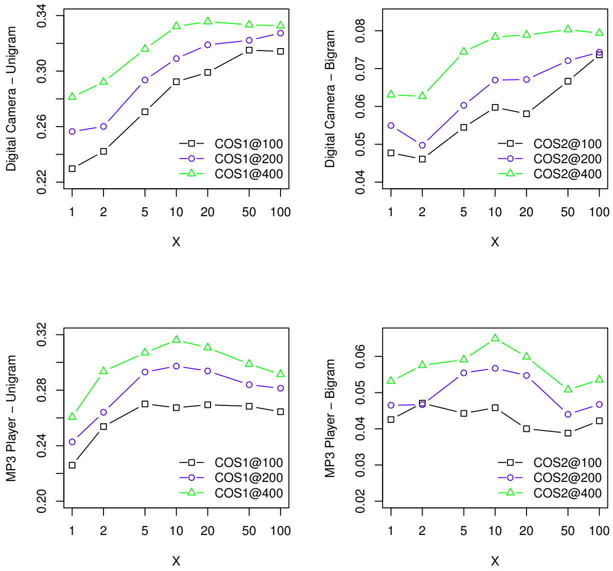


Figure 2: TFIDF cosine similarity evaluation results for Translation model with different number (X) of translating products. Upper figures are for digital cameras, and lower figures are for mp3 players. Left figures are results based on unigrams, and right figures are those base on bigrams.

We also evaluate retrieval results with ROUGE metrics. Although ROUGE does not consider importance of words, it is able to compute recall, precision, and F1 score in both unigram (ROUGE1-R, ROUGE1-P, and ROUGE1-F) and bigram (ROUGE2-R, ROUGE2-P, and ROUGE2-F) units. The ROUGE evaluation results for mp3 players are shown in Table 7. QL outperforms MEAD in all measures, but it is outperformed by MEAD-SIM in all measures since QL does not consider specifications similarity between products. SpecGen outperforms ReviewSpecGen in most measures (13/18) especially in bigram measures (9/9), which is different from the TFIDF cosine similarity results; this means that the sentences retrieved by SpecGen are more similar to ac-

Table 7: Unigram and bigram ROUGE @ K for MP3 Players category.

K	Model	ROUGE1-R	ROUGE1-P	ROUGE1-F	ROUGE2-R	ROUGE2-P	ROUGE2-F
100	QL	0.278	0.218	0.150	0.0545	0.0308	0.0297
	MEAD	0.202	0.217	0.132	0.0364	0.0242	0.0227
	MEAD-SIM	0.328	0.319	0.196	0.0615	0.0381	0.0367
	SpecGen	0.303	0.299	0.191	0.0727	0.0480	0.0406
	ReviewSpecGen	0.323	0.320	0.204	0.0650	0.0431	0.0378
	Translation	**0.369**†‡	**0.375**‡	**0.236**†‡	**0.1151**†‡	**0.0742**†‡	**0.0634**†‡
	(increase %)	(+11%)	(+12%)	(+20%)	(+87%)	(+95%)	(+73%)
200	QL	0.384	0.213	0.166	0.0812	0.0264	0.0300
	MEAD	0.266	0.171	0.127	0.0453	0.0186	0.0203
	MEAD-SIM	0.434	0.266	0.201	0.0834	0.0290	0.0333
	SpecGen	0.413	0.273	0.204	0.0913	0.0423	0.0395
	ReviewSpecGen	0.411	0.267	0.197	0.0848	0.0324	0.0344
	Translation	**0.481**†‡	**0.318**†‡	**0.239**†‡	**0.1582**†‡	**0.0664**†‡	**0.0657**†‡
	(increase %)	(+11%)	(+20%)	(+19%)	(+90%)	(+129%)	(+97%)
400	QL	0.501	0.186	0.175	0.1159	0.0205	0.0265
	MEAD	0.370	0.154	0.141	0.0517	0.0111	0.0146
	MEAD-SIM	0.560	0.224	0.210	0.1260	0.0207	0.0268
	SpecGen	0.535	0.221	0.207	0.1228	0.0293	0.0342
	ReviewSpecGen	0.546	0.221	0.204	0.1171	0.0254	0.0314
	Translation	**0.595**†‡	**0.240**†‡	**0.225**†‡	**0.2112**†‡	**0.0431**†‡	**0.0542**†‡
	(increase %)	(+6%)	(+7%)	(+7%)	(+68%)	(+108%)	(+102%)

tual reviews than those retrieved by ReviewSpecGen, but ReviewSpecGen retrieved more "important" relevant words. Translation model outperforms all other models in all measures (18/18), and the increase from MEAD-SIM and ReviewSpecGen is statistically significant in most measures (17/18 and 18/18, respectively). Similar to TFIDF cosine similarity results, the performance difference in bigram is clearer than in unigram, which means Translation model retrieves bigger fragments of actual reviews well. The increase in unigram ROUGE measures is not as big as that in unigram TFIDF cosine similarity measures, which means that the number of relevant words from Translation model is not very different from other models, but Translation model retrieves much more important relevant words.

We also evaluated retrieved sentences for digital cameras with ROUGE metrics. In general, Translation model outperforms other models in all measures. More specifically, it significantly outperforms MEAD-SIM and ReviewSpecGen in most measures (16/18 and 18/18, respectively). We do not list ROUGE evaluation results for digital cameras since the other patterns are similar to those for mp3 players.

Overall, ROUGE evaluation results are similar to cosine similarity evaluation results in general. The difference between the two metrics is that the TFIDF cosine similarity metric differentiates various models more clearly since they consider importance of word while the ROUGE metric does not; TFIDF cosine similarity metric prefers retrieved text that contains more important words, which is a desired property in such evaluation. On the other hand, ROUGE metric considers various evaluation aspects such as recall, precision, and F1 score, which can possibly help us analyze evaluation results in depth.

In order to reduce computation complexity of Translation model, we proposed to exploit X number of most promising products that are similar to the query product, instead of all products, under the assumption that similar products are likely to have similar reviews. We performed experiments with different X values to find how many translating products are needed to obtain reasonably good performance. The results are evaluated with TFIDF cosine similarity @ K for unigrams and bigrams, and the results are shown in Figure 2. Surprisingly, only a few translating products (e.g., ten) are enough to perform reasonably well especially for mp3 players. These results mean that only a few "good" translating products are enough to translate a candidate sentence well, and the "good" translating products may be selected by their similarity to the query product.

9. CONCLUSION AND FUTURE WORK

In this paper, we studied the problem of automatic relevant review text retrieval for products having no reviews. Relevant review sentences for new or unpopular products can be very useful for consumers who seek for relevant opinions, but no previous work has addressed this novel problem. We proposed several methods to solve this problem, including summarization-based methods such as MEAD and MEAD-SIM and probabilistic retrieval methods such as Specifications Generation model, Review and Specifications Generation model, and Translation model. To evaluate relevance of retrieved opinion sentences in the situation where human-labeled judgments are not available, we measured the proximity between the retrieved text and the actual reviews of a query product. Experiment results show that our novel Translation model indeed retrieves useful sentences and significantly outperforms the baseline methods.

Our work opens up a new direction in text data mining and opinion analysis. The new problem of review text retrieval for new products can be studied from multiple perspectives. First, it can be regarded as a summarization problem as the retrieved sentences need to be central across different reviews. Second, as done in this paper, it can also be regarded as a special retrieval problem with the goal of retrieving relevant opinions with product specifications as a query. Finally, it can also be studied from the perspective of collaborative filtering where we would leverage related products to recommend relevant "opinions" to new products. All these are interesting future directions that can potentially lead to even more accurate and more useful algorithms.

10. ACKNOWLEDGMENTS

This work is supported in part by a gift fund from TCL and by the National Science Foundation under Grant Number CNS-1027965.

11. REFERENCES

[1] A. Berger and J. Lafferty. Information retrieval as statistical translation. In *Proceedings of ACM SIGIR 1999*, pages 222–229, 1999.

[2] I. Bhattacharya, S. Godbole, and S. Joshi. Structured entity identification and document categorization: two tasks with one joint model. In *Proceedings of ACM KDD 2008*, pages 25–33, 2008.

[3] C. Dellarocas, X. M. Zhang, and N. F. Awad. Exploring the value of online product reviews in forecasting sales: The case of motion pictures. *Journal of Interactive marketing*, 21(4):23–45, 2007.

[4] H. Duan, C. Zhai, J. Cheng, and A. Gattani. Supporting keyword search in product database: A probabilistic approach. *Proc. VLDB Endow.*, 6(14):1786–1797, Sept. 2013.

[5] G. Ganu, N. Elhadad, and A. Marian. Beyond the stars: Improving rating predictions using review text content.

[6] E. Hovy and C.-Y. Lin. Automated text summarization in SUMMARIST. In I. Mani and M. T. Maybury, editors, *Advances in Automatic Text Summarization*. MIT Press, 1999.

[7] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proceedings of KDD '04*, pages 168–177, 2004.

[8] F. Jelinek. Interpolated estimation of markov source parameters from sparse data. *Pattern recognition in practice*, 1980.

[9] H. D. Kim, K. Ganesan, P. Sondhi, and C. Zhai. Comprehensive review of opinion summarization. *Computer Science Research and Tech Reports*, 2011.

[10] H. D. Kim and C. Zhai. Generating comparative summaries of contradictory opinions in text. In *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*, pages 385–394, New York, NY, USA, 2009. ACM.

[11] J. Kupiec, J. Pedersen, and F. Chen. A trainable document summarizer. In *Proceedings of ACM SIGIR '95*, pages 68–73, 1995.

[12] M. Lalmas. Xml retrieval (synthesis lectures on information concepts, retrieval, and services). *Morgan and Claypool, San Rafael, CA*, 2009.

[13] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, 2004.

[14] B. Liu. Sentiment analysis and subjectivity. In N. Indurkhya and F. J. Damerau, editors, *Handbook of Natural Language Processing, Second Edition*. CRC Press, Taylor and Francis Group, Boca Raton, FL, 2010. ISBN 978-1420085921.

[15] B. Liu, M. Hu, and J. Cheng. Opinion observer: analyzing and comparing opinions on the web. In *Proceedings of WWW '05*, pages 342–351, 2005.

[16] Y. Lu, C. Zhai, and N. Sundaresan. Rated aspect summarization of short comments. In *Proceedings of WWW '09*, pages 131–140, 2009.

[17] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, 2014.

[18] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of WWW '07*, pages 171–180, 2007.

[19] C. D. Paice. Constructing literature abstracts by computer: techniques and prospects. *Inf. Process. Manage.*, 26(1):171–186, 1990.

[20] B. Pang and L. Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 115–124. Association for Computational Linguistics, 2005.

[21] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135, 2008.

[22] D. H. Park, C. Zhai, and L. Guo. Speclda: Modeling product reviews and specifications to generate augmented specifications. In *Proceedings of the 2015 SIAM International Conference on Data Mining*. SIAM, 2015.

[23] I. Peñalver-Martínez, R. Valencia-García, and F. García-Sánchez. Ontology-guided approach to feature-based opinion mining. In *Natural Language Processing and Information Systems*, pages 193–200. Springer, 2011.

[24] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281. ACM, 1998.

[25] A.-M. Popescu and O. Etzioni. Extracting product features and opinions from reviews. In *Proceedings of HLT-EMNLP 2005*, pages 339–346, 2005.

[26] D. R. Radev, H. Jing, and M. Budzikowska. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *Proceedings of the 2000 NAACL-ANLP Workshop on Automatic Summarization*, pages 21–30. Association for Computational Linguistics, 2000.

[27] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260. ACM, 2002.

[28] I. Titov and R. McDonald. Modeling online reviews with multi-grain topic models. In *Proceedings of WWW '08*, pages 111–120, 2008.

[29] T. Wang, Y. Cai, G. Zhang, Y. Liu, J. Chen, and H. Min. Product feature summarization by incorporating domain information. In *Database Systems for Advanced Applications*. Springer, 2013.

[30] J. Yu, Z.-J. Zha, M. Wang, K. Wang, and T.-S. Chua. Domain-assisted product aspect hierarchy generation: towards hierarchical organization of unstructured consumer reviews. In *Proceedings of EMNLP 2011*, pages 140–150, 2011.

[31] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 334–342. ACM, 2001.

[32] L. Zhou and P. Chaovalit. Ontology-supported polarity mining. *Journal of the American Society for Information Science and technology*, 59(1):98–110, 2008.